

Package: trama.documentos (via r-universe)

October 20, 2026

Title Living Documents for trama Analyses

Version 0.4.0

Description Turns a trama analysis into a finished document: a Quarto file of the project where the text belongs to the author and managed regions belong to trama. Results of flow nodes are materialized next to the document (tables, figures, sanitized models, tests) and referenced by regions that are refreshed when the analysis changes, without ever touching the author's text. Documents start from Quarto extension templates (a gallery to create, edit, duplicate, import and export them, with built-in academic ABNT, consulting report, article, slides and dashboard templates) and author profiles, and render locally to HTML, PDF (Typst), Word, revealjs slides and dashboards. Charts follow a light theme derived from the template's brand. A web page template turns models into predictors that run in the browser through WebR, and any document can be published to GitHub as an RO-Crate research package, with a release and a GitHub Pages site rendered by GitHub Actions. It is a trama plugin, not a block collection: it has no nodes. When installed, the trama editor shows a Documents panel (with a template gallery) and an 'insert into document' item in the card menu.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1)

Imports trama (>= 0.5.25), trama.data (>= 0.4.7), trama.github (>= 0.2.0), curl, digest, ggplot2, httpuv, jsonlite, later, processx, ragg, rlang, shiny, stats, tools, utils, yaml, zip

Suggests testthat (>= 3.0.0), withr, svglite, png, trama.view, trama.models

Config/testthat/edition 3

Config/trama/plugin true

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0
Repository https://uaipetro.r-universe.dev
Date/Publication 2026-10-19 22:43:33 UTC
RemoteUrl https://github.com/uaipetro/trama
RemoteRef HEAD
RemoteSha d6636116b9f41a64105380c34b766826e132f0c7
RemoteSubdir plugins/trama.documentos

Contents

tr_documento_acompanhar	2
tr_documento_atualizar	3
tr_documento_componente	4
tr_documento_criar	4
tr_documento_estado	5
tr_documento_formatos	6
tr_documento_inserir	7
tr_documento_limpar	8
tr_documento_listar	8
tr_documento_migrar_publicar	9
tr_documento_modelo	9
tr_documento_modelo_criar	10
tr_documento_modelo_exportar	12
tr_documento_modelos	12
tr_documento_pacote	13
tr_documento_parte	14
tr_documento_perfis	15
tr_documento_previa	16
tr_documento_publicacao	16
tr_documento_publicacoes	17
tr_documento_publicar	18
tr_documento_renderizar	19
tr_documento_resolver	19
trama_plugin	20

Index	21
--------------	-----------

tr_documento_acompanhar
<i>Acompanha a última publicação do documento</i>

Description

Um passo de leitura no GitHub: o workflow começou, terminou, falhou (em que etapa)? A release saiu? A página responde? Grava e devolve o estado, com etapa em enviado, actions, release, no_ar ou falhou.

Usage

```
tr_documento_acompanhar(documento, projeto = NULL)
```

Arguments

documento	Caminho do .qmd.
projeto	Um trama::tr_project() ou a pasta dele; NULL acha a primeira pasta com trama.json acima do documento.

Value

O estado do envio (ver `tr_documento_publicar()`), com actions, release e log quando houver; NULL se o documento nunca foi publicado deste projeto.

tr_documento_atualizar

Atualiza as regiões cujo resultado mudou

Description

Para cada região desatualizada (ou todas, com forcar), recalcula o valor, materializa e compara pelo hash do resultado: igual, nada é reescrito; diferente, reescreve a pasta _trama/<chave>/ e, se o corpo da região mudou, só as linhas dela. Regiões em conflito e órfãs não são tocadas. Antes de gravar, confere que o texto de fora das regiões ficou idêntico, byte a byte.

Usage

```
tr_documento_atualizar(  
  documento,  
  projeto = NULL,  
  regioes = NULL,  
  forcar = FALSE  
)
```

Arguments

documento	Caminho do .qmd.
projeto	Um trama::tr_project() ou a pasta dele; NULL acha a primeira pasta com trama.json acima do documento.
regioes	Chaves das regiões a atualizar; NULL, todas.
forcar	TRUE recalcula também as que parecem atualizadas.

Value

data.frame com regioao, estado (atualizada, igual, conflito, orfa, indisponivel, erro) e mensagem.

tr_documento_componente

Materializa um resultado para o documento

Description

Genérico S3 pela classe do valor. Métodos embutidos: tabela (data.frame), gráfico (ggplot), modelo (tr_models_fit, reduzido por trama.models::tr_models_strip()) e teste (tr_test). Para os demais, o método padrão usa o report do tipo da porta (trama::tr_type(report =)), o mesmo que o relatório exportado usa; sem ele, a parte sai com erro.

Usage

```
tr_documento_componente(x, ctx)
```

Arguments

x	O valor do nó.
ctx	Lista com no, porta, chave (pasta em _trama/), titulo (o rótulo do card, ou o do bloco), tipo (o tr_type da porta, ou NULL), opcoes (atributos trama-* da região: linhas, formato, largura, altura), web (TRUE quando o documento roda R no navegador, como o modelo Página web: um modelo vira preditor) e arquivo(nome), que dá o caminho de um arquivo da parte relativo ao documento.

Details

Uma coleção estende registrando um método para a classe dos seus valores (S3method(trama.documentos::tr_documento, minha_classe), ou registerS3method() no .onLoad, se o plugin for só sugerido).

Value

Um [tr_documento_parte\(\)](#).

tr_documento_criar

Cria um documento a partir de um modelo e de um perfil

Description

Monta documentos/<nome>/ no projeto: o .qmd (YAML com título, autor do perfil, data, formatos do modelo e o fluxo de origem, mais o esqueleto do modelo), a cópia da extensão em _extensions/<modelo>/, o _brand.yml e um _quarto.yml que manda a saída do render para _saida/. É o único momento em que o trama escreve fora das regiões.

Usage

```
tr_documento_criar(
  nome,
  projeto = ".",
  modelo = "basico",
  perfil = NULL,
  titulo = NULL,
  fluxo = "main"
)
```

Arguments

nome	Nome do documento (letras, dígitos, -, _): pasta e arquivo .qmd.
projeto	Pasta do projeto (ou um trama::tr_project()).
modelo	Nome do modelo (tr_documento_modelos()).
perfil	Nome do perfil de autor; NULL, o que o modelo sugere (se o perfil existe) ou o padrão (se houver); FALSE, sem autor.
titulo	Título; NULL usa o nome.
fluxo	Fluxo do projeto de onde vêm as regiões.

Details

O YAML do esqueleto pode pedir campos do perfil: um valor "{{perfil.orientador}}" vira o orientador do perfil (e some, se o perfil não tem), e "{{ano}}" vira o ano corrente. É assim que o modelo UFLA preenche orientador e curso.

Value

O caminho do .qmd.

tr_documento_estado	<i>Estado das regiões do documento</i>
---------------------	--

Description

Não calcula nada: compara o conteúdo de cada região com o hash que o trama gravou e a chave do store de agora com a da última materialização.

Usage

```
tr_documento_estado(documento, projeto = NULL)
```

Arguments

documento	Caminho do .qmd.
projeto	Um trama::tr_project() ou a pasta dele; NULL acha a primeira pasta com trama.json acima do documento.

Details

- atualizada: a região mostra o resultado de agora;
- desatualizada: o nó, o que ele lê ou as opções mudaram desde então (atualizar confirma pelo resultado e só reescreve se ele mudou);
- conflito: o conteúdo foi editado à mão; atualizar não toca nela;
- orfa: o nó (ou a porta) saiu do fluxo; a região fica até a pessoa decidir;
- indisponível: o nó está no fluxo, mas a coleção dele não carregou;
- duplicada: outra região com a mesma chave veio antes no texto (cópia à mão); só a primeira é do trama, e esta não é tocada.

Value

data.frame com regioao, no, porta, estado e linha, na ordem do texto.

tr_documento_formatos *Formatos de saída do documento*

Description

Os formatos que o YAML do documento declara (vindos do modelo), nos nomes de `tr_documento_renderizar()`.

Usage

```
tr_documento_formatos(documento)
```

Arguments

documento Caminho do .qmd.

Value

Vetor de texto, como `c("html", "pdf", "docx")`.

tr_documento_inserir *Insere no documento a região de um nó*

Description

Calcula (ou lê do cache) o valor do nó, materializa em `_trama/<no>/` ao lado do documento e insere a região `::: {#trama-<no> .trama}` que o referencia. Só acrescenta linhas: o texto existente não muda.

Usage

```
tr_documento_inserir(
  documento,
  no,
  projeto = NULL,
  porta = NULL,
  apos = NULL,
  opcoes = list(),
  aceitar_dados_treino = FALSE
)
```

Arguments

documento	Caminho do .qmd.
no	Id do nó no fluxo do documento (YAML trama: <code>fluxo:</code> , padrão "main").
projeto	Um <code>trama::tr_project()</code> ou a pasta dele; NULL acha a primeira pasta com <code>trama.json</code> acima do documento.
porta	Porta de saída; NULL é a primeira.
apos	Onde inserir: NULL (fim do documento), o número da linha depois da qual entra (0, logo depois do YAML) ou a chave de uma região existente (entra depois dela). Linha dentro do YAML, de região ou de bloco de código é recusada.
opcoes	Lista de opções da região, gravadas no cabeçalho como <code>trama-<opção></code> (a pessoa pode mudá-las ali): <code>linhas</code> (tabela), <code>tabela = "completa"</code> (o <code>dados.csv</code> leva a tabela inteira, e não só as linhas mostradas), <code>formato</code> ("png" ou "svg"), <code>largura</code> e <code>altura</code> (figura, polegadas).
aceitar_dados_treino	TRUE aceita pôr no documento um modelo que carrega linhas do treino (SVM, floresta de regressão...). O aceite fica no cabeçalho da região (<code>trama-dados-treino="aceito"</code>).

Value

Invisível: lista com `regiao` (a chave) e avisos.

tr_documento_limpar	<i>Apaga de _trama/ o que nenhuma região usa</i>
---------------------	--

Description

Uma pasta fica se há região com a chave dela ou se o texto ainda cita o caminho (região mantida como texto da pessoa), em qualquer .qmd da pasta do documento.

Usage

```
tr_documento_limpar(documento)
```

Arguments

documento	Caminho do .qmd.
-----------	------------------

Value

Invisível: as pastas apagadas.

tr_documento_listar	<i>Documentos do projeto</i>
---------------------	------------------------------

Description

Documentos do projeto

Usage

```
tr_documento_listar(projeto = ".")
```

Arguments

projeto	Pasta do projeto (ou um trama::tr_project()).
---------	---

Value

data.frame com nome, caminho, fluxo, modelo e regioes.

`tr_documento_migrar_publicar`*Migra os nós publicar/publish do projeto para documentos*

Description

Para cada nó publicar/publish (da coleção trama.publicar, que deixou de existir) em qualquer fluxo do projeto, cria um documento do modelo Página web com o que o nó tinha: título, resumo, palavras-chave, autores (com ORCID), licenças, tipo de versão, repositório privado e aceite de dados de treino no YAML; o corpo segue o layout do nó (cabeçalho, textos, cada item ligado como região, o diagrama). O endereço da publicação já feita passa a ser do documento (`tr_documento_publicacoes()`), e a próxima publicação continua no mesmo repositório, na versão seguinte. Depois tira o nó (e as arestas que chegavam nele) do fluxo e trama.publicar da lista de coleções do trama.json.

Usage

```
tr_documento_migrar_publicar(projeto = ".")
```

Arguments

projeto Pasta do projeto (ou um trama::tr_project()).

Details

Item que não pôde virar região (o nó falha, o modelo carrega dados de treino sem o aceite) vira uma nota no texto, para inserir de novo à mão.

Value

data.frame com fluxo, no, documento (caminho relativo ao projeto), regioes (quantas entraram) e avisos; zero linhas se não havia nada a migrar.

`tr_documento_modelo`*Um modelo de documento, em detalhe*

Description

O que a galeria mostra e edita: título, descrição, formatos, cores, fontes, logo, capa, CSL e o perfil sugerido, mais se o modelo é válido.

Usage

```
tr_documento_modelo(nome)
```

Arguments

nome Nome do modelo (a pasta), como em `tr_documento_modelos()`.

Value

Lista com nome, titulo, descricao, formatos, formatos_desligados, origem ("embutido" ou "usuario"), sobre_embutido (cópia do usuário de um embutido), caminho, perfil, cores e fontes (listas nomeadas como em `tr_documento_modelo_editar()`), logo (caminho ou NULL), capa (TRUE, FALSE ou NULL se o modelo não diz), csl (nome do arquivo ou NULL), valido e problemas.

`tr_documento_modelo_criar`

Cria, edita, duplica e remove modelos do usuário

Description

Os modelos do usuário moram em `tools::R_user_dir("trama", "data")/modelos/`. `tr_documento_modelo_criar()` parte de um modelo existente (base) e aplica as edições; `tr_documento_modelo_duplicar()` copia tal como está; `tr_documento_modelo_editar()` muda os campos dados (os demais ficam); `tr_documento_modelo_remover()` apaga um modelo do usuário (o embutido de mesmo nome, se houver, volta a valer).

Usage

```
tr_documento_modelo_criar(
  nome,
  titulo = NULL,
  base = "basico",
  descricao = NULL,
  formatos = NULL,
  cores = NULL,
  fontes = NULL,
  logo = NULL,
  capa = NULL,
  csl = NULL,
  perfil = NULL
)

tr_documento_modelo_duplicar(modelo, nome, titulo = NULL)

tr_documento_modelo_editar(
  nome,
  titulo = NULL,
  descricao = NULL,
  formatos = NULL,
  cores = NULL,
```

```

fontes = NULL,
logo = NULL,
capa = NULL,
csl = NULL,
perfil = NULL
)

tr_documento_modelo_remove(nome)

```

Arguments

nome	Nome do modelo: letras minúsculas, dígitos e - (é a pasta e o prefixo do formato no Quarto, <nome>-html).
titulo	Nome de exibição na galeria ("UFLA (ABNT)").
base	Modelo de onde tr_documento_modelo_criar() parte.
descricao	Uma frase sobre o modelo.
formatos	Formatos habilitados, entre "html", "typst" (PDF), "docx", "revealjs" (slides) e "dashboard". Desligar guarda a configuração do formato; religar a traz de volta.
cores	Lista ou vetor nomeado de cores #rrggbb: primaria, secundaria, destaque, fundo, texto (vão para o _brand.yml). Os gráficos do documento usam estas cores num tema claro.
fontes	Lista ou vetor nomeado de famílias: texto, titulos, codigo. A fonte precisa estar instalada para o PDF.
logo	Caminho de uma imagem (png, svg, jpg), copiada para o modelo; FALSE tira o logo.
capa	TRUE ou FALSE: capa nos modelos que têm (UFLA, Consultoria). Vai como metadado capa do formato.
csl	Caminho de um estilo de citação .csl, copiado para o modelo; FALSE volta ao estilo padrão do Quarto.
perfil	Perfil de autor que o modelo sugere (tr_documento_perfis()); FALSE tira a sugestão.
modelo	Modelo a duplicar.

Details

Editar um embutido cria a cópia do usuário com o mesmo nome, que passa a valer em documentos novos. Documento já criado leva a cópia do modelo de quando foi criado: mudar o modelo não muda documento existente.

Value

Invisível: o modelo, como tr_documento_modelo(); remove devolve o nome.

tr_documento_modelo_exportar

Exporta e importa um modelo como zip

Description

O zip leva a pasta do modelo (<nome>/_extension.yml, ...). Importar aceita esse formato, a pasta solta na raiz do zip e o formato das extensões Quarto publicadas (_extensions/<nome>/ com o template.qmd na raiz). O modelo é conferido antes de entrar: zip inválido não deixa nada na galeria.

Usage

```
tr_documento_modelo_exportar(nome, arquivo = NULL)
```

```
tr_documento_modelo_importar(arquivo, nome = NULL, substituir = FALSE)
```

Arguments

nome	Nome do modelo. Na importação, NULL usa o da pasta do zip.
arquivo	Caminho do zip. Na exportação, NULL grava <nome>.zip na pasta atual.
substituir	Importar por cima de um modelo do usuário de mesmo nome.

Value

O caminho do zip (exportar) ou, invisível, o modelo importado ([tr_documento_modelo\(\)](#)).

tr_documento_modelos *Modelos de documento disponíveis*

Description

A galeria: os embutidos e os do usuário (que têm precedência sobre o embutido de mesmo nome). Modelo inválido aparece, com valido = FALSE e o problema, em vez de sumir ou quebrar a lista.

Usage

```
tr_documento_modelos()
```

Value

data.frame com nome, titulo, formatos (separados por vírgula), origem ("embutido" ou "usuario"), caminho, descricao, perfil (o perfil de autor que o modelo sugere, ou NA), sobre_embutido, valido e problema (os problemas, separados por "; ", ou NA).

See Also

[tr_documento_modelo\(\)](#) para um modelo em detalhe e [tr_documento_modelo_criar\(\)](#) para criar, editar, duplicar e remover.

tr_documento_pacote	<i>Monta o pacote de publicação de um documento</i>
---------------------	---

Description

A pasta que "Publicar no GitHub" envia, e que se pode conferir antes: o documento como `index.qmd`, a pasta dele (sem `_saida/` nem arquivos ocultos; de `_trama/`, só o que o texto usa), `flow.json`, `fluxo.svg`, `dados-entrada.json`, `ambiente.json`, `README.md`, `CITATION.cff`, `LICENSE`, `_variables.yml` com a versão e a citação, e o `ro-crate-metadata.json`. Bloqueios de [tr_documento_publicacao\(\)](#) param a montagem.

Usage

```
tr_documento_pacote(  
  documento,  
  dir,  
  projeto = NULL,  
  versao = NULL,  
  repositorio = "",  
  data = Sys.Date()  
)
```

Arguments

documento	Caminho do <code>.qmd</code> .
dir	Pasta de destino (criada; o conteúdo anterior é apagado).
projeto	Um <code>trama::tr_project()</code> ou a pasta dele; <code>NULL</code> acha a primeira pasta com <code>trama.json</code> acima do documento.
versao	Versão do pacote (<code>"1.2.0"</code>); <code>NULL</code> , <code>"rascunho"</code> .
repositorio	Endereço do repositório da publicação (vai para a citação).
data	Data da publicação (<code>Date</code>).

Value

Invisível: `list(dir, arquivos, publicacao)`.

tr_documento_parte	<i>Parte de documento: o que um método de tr_documento_componente() devolve</i>
--------------------	---

Description

O motor grava arquivos em `_trama/<chave>/` ao lado do documento e põe corpo dentro da região. O corpo referencia os arquivos pelo caminho relativo ao documento (`ctx$arquivo("tabela.md")`), para que o render não precise recalculá-los nada nem ter o store do projeto.

Usage

```
tr_documento_parte(
  tipo,
  titulo = "",
  arquivos = list(),
  corpo = character(),
  metadados = list(),
  dados_treino = NULL,
  avisos = character(),
  erro = NULL
)
```

Arguments

tipo	Texto curto do que é a parte ("tabela", "figura", "modelo", "teste", ou o que a coleção quiser).
titulo	Título da parte (legenda de figura e tabela).
arquivos	Lista nomeada: nome do arquivo (sem / nem .) -> conteúdo em raw ou texto.
corpo	Linhas de Markdown que vão dentro da região.
metadados	Lista que descreve a parte no pacote RO-Crate: tipo_schema, descricao, propriedades (pares JSON-LD).
dados_treino	NULL, ou texto dizendo que linhas do treino a parte carrega (a marca de trama.models::tr_models_stri). Parte marcada só entra no documento com aceite explícito.
avisos	Textos para quem insere (não bloqueiam).
erro	Texto: a parte não entra.

Value

Objeto de classe `tr_documento_parte`.

tr_documento_perfis	<i>Perfis de autor</i>
---------------------	------------------------

Description

tr_documento_perfis() lê todos; tr_documento_perfil() devolve um (o padrão, sem nome); tr_documento_perfil_salvar() cria ou substitui; tr_documento_perfil_remove() apaga. Ficam em `tools::R_user_dir("trama", "data")/perfis.yml`.

Usage

```
tr_documento_perfis()

tr_documento_perfil(perfil = NULL)

tr_documento_perfil_salvar(
  perfil,
  nome,
  email = NULL,
  orcid = NULL,
  instituicao = NULL,
  curso = NULL,
  orientador = NULL,
  logo = NULL,
  padrao = FALSE
)

tr_documento_perfil_remove(perfil)
```

Arguments

perfil	Nome do perfil (letras, dígitos, -, _), como "aluno-ufla".
nome	Nome da pessoa, como sai no documento.
email, orcid, instituicao, curso, orientador, logo	Opcionais. O ORCID é conferido (formato e dígito verificador). curso vale também para cargo. logo é o caminho de uma imagem.
padrao	TRUE torna este o perfil padrão. O primeiro perfil salvo vira padrão sozinho.

Value

tr_documento_perfis(): lista com padrao (nome ou NULL) e perfis (lista nomeada). tr_documento_perfil(): o perfil (lista com perfil e os campos) ou NULL. As demais, invisível, o perfil salvo ou o nome removido.

tr_documento_previa	<i>Prévia do documento no navegador</i>
---------------------	---

Description

Renderiza em HTML (ou nos slides ou no dashboard, se o documento não tem HTML) e serve _saida/ num servidor local (<http://127.0.0.1:<porta>/>). O servidor fica de pé até tr_documento_previa_parar() ou o fim da sessão do R; uma nova prévia do mesmo documento reaproveita o servidor.

Usage

```
tr_documento_previa(documento, abrir = interactive(), tempo = 300)
```

```
tr_documento_previa_parar(previa = NULL)
```

Arguments

documento	Caminho do .qmd.
abrir	Abrir no navegador.
tempo	Segundos até desistir.
previa	Uma prévia; NULL para todas. tr_documento_previa_parar() devolve, invisível, quantos servidores parou.

Value

Objeto tr_documento_previa: lista com url, pasta e documento.

tr_documento_publicacao	<i>O que sairia na publicação de um documento</i>
-------------------------	---

Description

Não toca no GitHub nem grava nada: lê o YAML e o estado das regiões e diz o que vai no pacote, o que merece atenção (avisos) e o que impede a publicação (erros).

Usage

```
tr_documento_publicacao(
  documento,
  projeto = NULL,
  desatualizada = c("erro", "aviso")
)
```

Arguments

documento	Caminho do .qmd.
projeto	Um trama::tr_project() ou a pasta dele; NULL acha a primeira pasta com trama.json acima do documento.
desatualizada	Região desatualizada é "erro" (o padrão: o pacote não sai) ou "aviso" (a conferência antes de <code>tr_documento_publicar()</code> , que atualiza o documento antes de montar).

Details

Do YAML do documento: title, author (nomes e, se houver, orcid, conferido pelo dígito verificador), description (ou abstract) e keywords. De trama: publicar:: licenca (do conteúdo, padrão CC-BY-4.0), licenca-codigo (padrão MIT), versao (tipo da próxima: menor, correção ou maior), privado (repositório da publicação privado; Pages privado exige plano pago) e aceitar-dados-treino (publicar modelos que carregam linhas do treino, que de outro modo são recusados) e incluir-dados (arquivos de dados que estejam na pasta do documento).

Bloqueiam: falta de título ou de autor, ORCID que não confere, licença desconhecida, região desatualizada (atualize antes), região sem a pasta materializada, modelo com dados de treino sem o aceite, arquivo acima de 100 MB, documento sem formato HTML ou com mais de um (a página do GitHub Pages), arquivo de dados na pasta do documento sem incluir-dados e arquivo da pasta com o nome de um que o pacote gera. Os outros arquivos da pasta, que vão junto, saem nos avisos.

Value

Objeto tr_documento_publicacao: lista com documento, metadados, regioes, avisos e erros.

tr_documento_publicacoes

Publicações do projeto

Description

Em que repositório cada documento é publicado, a página, a última versão enviada e quando. Mora no trama.json (chave publicacoes, pelo caminho do documento no projeto): é configuração do PROJETO, não vai no documento nem no pacote publicado. Estado do Actions e versões ao vivo se leem do GitHub (trama.github); isto é só o endereço de cada publicação.

Usage

```
tr_documento_publicacoes(projeto = ".")
```

Arguments

projeto	Pasta do projeto (ou um trama::tr_project()).
---------	---

Value

data.frame com documento (caminho relativo ao projeto), repo, url, pagina (base do Pages), ultimo, versao, tag e data; zero linhas sem publicação.

tr_documento_publicar *Publica o documento no GitHub*

Description

Atualiza as regiões, confere ([tr_documento_publicacao\(\)](#)), monta o pacote ([tr_documento_pacote\(\)](#)) com a próxima versão livre e o envia num commit atômico ao repositório da publicação, com a tag vX.Y.Z que dispara o workflow. A conta é a do trama.github (menu GitHub do editor, ou GITHUB_PAT). Volta logo: o Actions leva um minuto ou dois, e [tr_documento Acompanhar\(\)](#) diz em que etapa está.

Usage

```
tr_documento_publicar(
  documento,
  projeto = NULL,
  repositorio = NULL,
  confirmado = FALSE,
  atualizar = TRUE
)
```

Arguments

documento	Caminho do .qmd.
projeto	Um trama::tr_project() ou a pasta dele; NULL acha a primeira pasta com trama.json acima do documento.
repositorio	"nome" ou "dono/nome"; NULL usa o que o documento já usou (registro do projeto) ou o título.
confirmado	Já confirmado (ver acima).
atualizar	Atualizar as regiões antes (tr_documento_atualizar()).

Details

Pede confirmação (devolve confirmar, sem tocar no GitHub) antes de criar um repositório privado (Pages privado exige plano pago) e antes de enviar a um repositório que já existe sem nenhuma versão vX.Y.Z (o envio substitui o conteúdo dele). Bloqueios saem em bloqueios.

Value

Lista: bloqueios, ou confirmar (texto) e repo, ou o estado do envio (documento, repo, url, versao, tag, sha, pagina, ultimo, etapa = "enviado", detalhe).

tr_documento_renderizar

Renderiza o documento com o Quarto

Description

Usa o formato do modelo que o YAML do documento declara (basico-typst para "pdf", por exemplo); sem ele, o formato puro do Quarto. Não atualiza regiões: renderiza o que está no arquivo (chame `tr_documento_atualizar()` antes, se quiser os resultados de agora).

Usage

```
tr_documento_renderizar(
  documento,
  formato = c("html", "pdf", "docx", "revealjs", "dashboard"),
  tempo = 300
)
```

Arguments

documento	Caminho do .qmd.
formato	"html", "pdf" (via Typst), "docx", "revealjs" (slides) ou "dashboard". Os que o documento declara: <code>tr_documento_formatos()</code> .
tempo	Segundos até desistir.

Value

O caminho do arquivo gerado, em `_saida/` (slides em `_saida/slides/`, dashboard em `_saida/painel/`).

tr_documento_resolver *Resolve uma região em conflito ou órfã*

Description

- manter: a versão que está no texto passa a ser da pessoa: as cercas da região saem e o conteúdo fica como texto comum, que o trama não toca mais. A pasta `_trama/<chave>/` é congelada como `_trama/<chave>--mantida/` (e o texto passa a citá-la), para que um novo "inserir" do mesmo nó não mude o que a pessoa ficou;
- trazer: descarta a edição e põe a versão do trama de agora (não vale para órfã);
- remover: apaga a região do texto.

Usage

```
tr_documento_resolver(
  documento,
  regioao,
  acao = c("manter", "trazer", "remover"),
  projeto = NULL,
  confirmar = FALSE
)
```

Arguments

documento	Caminho do .qmd.
regiao	Chave da região ("lm" para #trama-lm).
acao	"manter", "trazer" ou "remover".
projeto	Um trama::tr_project() ou a pasta dele; NULL acha a primeira pasta com trama.json acima do documento.
confirmar	Ver acima.

Details

Região em conflito com cercas ::: dentro (a pessoa pôs um callout, por exemplo) pode ter o fim diferente do que a pessoa imagina: trazer e remover apagarão essas linhas, e só seguem com confirmar = TRUE.

Value

Invisível: lista com regioao e removidas, as linhas que saíram do texto.

trama_plugin	<i>Painel Documentos no editor</i>
--------------	------------------------------------

Description

Contrato de plugin do trama (ver trama::tr_plugins()): o editor carrega inst/trama/documentos.js (o botão Documentos na barra e o item "Inserir no documento..." no menu do card) e liga o server abaixo a cada sessão. O canal é próprio: o JS manda input\$trama_documentos ({acao, ...}) e o R responde com a mensagem trama_documentos ({tipo, ...}).

Usage

```
trama_plugin()
```

Value

Lista com js, css e server.

Index

tr_documento_acompanhar, [2](#)
tr_documento_acompanhar(), [18](#)
tr_documento_atualizar, [3](#)
tr_documento_atualizar(), [18](#), [19](#)
tr_documento_componente, [4](#)
tr_documento_criar, [4](#)
tr_documento_estado, [5](#)
tr_documento_formatos, [6](#)
tr_documento_formatos(), [19](#)
tr_documento_inserir, [7](#)
tr_documento_limpar, [8](#)
tr_documento_listar, [8](#)
tr_documento_migrar_publicar, [9](#)
tr_documento_modelo, [9](#)
tr_documento_modelo(), [11–13](#)
tr_documento_modelo_criar, [10](#)
tr_documento_modelo_criar(), [13](#)
tr_documento_modelo_duplicar
 (tr_documento_modelo_criar), [10](#)
tr_documento_modelo_editar
 (tr_documento_modelo_criar), [10](#)
tr_documento_modelo_editar(), [10](#)
tr_documento_modelo_exportar, [12](#)
tr_documento_modelo_importar
 (tr_documento_modelo_exportar),
 [12](#)
tr_documento_modelo_remove
 (tr_documento_modelo_criar), [10](#)
tr_documento_modelos, [12](#)
tr_documento_modelos(), [5](#), [10](#)
tr_documento_pacote, [13](#)
tr_documento_pacote(), [18](#)
tr_documento_parte, [14](#)
tr_documento_parte(), [4](#)
tr_documento_perfil
 (tr_documento_perfis), [15](#)
tr_documento_perfil_remove
 (tr_documento_perfis), [15](#)
tr_documento_perfil_salvar
 (tr_documento_perfis), [15](#)
tr_documento_perfis, [15](#)
tr_documento_perfis(), [11](#)
tr_documento_previa, [16](#)
tr_documento_previa_parar
 (tr_documento_previa), [16](#)
tr_documento_publicacao, [16](#)
tr_documento_publicacao(), [13](#), [18](#)
tr_documento_publicacoes, [17](#)
tr_documento_publicacoes(), [9](#)
tr_documento_publicar, [18](#)
tr_documento_publicar(), [3](#), [17](#)
tr_documento_renderizar, [19](#)
tr_documento_renderizar(), [6](#)
tr_documento_resolver, [19](#)
trama_plugin, [20](#)